Proactive Distributed Constraint Optimization Problems

Doctoral Mentoring Program

Khoi Hoang
Washington University in St. Louis
St. Louis, MO
khoi.hoang@wustl.edu

ABSTRACT

Current approaches that model dynamism in DCOPs solve a sequence of static problems, *reacting* to changes in the environment as the agents observe them. Such approaches thus ignore possible predictions on future changes. To overcome this limitation, we introduce (finite-horizon) *Proactive Dynamic DCOPs* (*PD-DCOPs*) and *Infinite-Horizon PD-DCOPs* (*IPD-DCOPs*) to model dynamic DCOPs in the presence of exogenous uncertainty. In contrast to reactive approaches, PD-DCOPs and IPD-DCOPs are able to explicitly model the possible changes to the problem, and take such information into account *proactively*, when solving the dynamically changing problem. The additional expressivity of these formalisms allows them to model a wider variety of distributed optimization problems. Our work presents both theoretical and practical contributions that advance current dynamic DCOP models.

KEYWORDS

Distributed Problem Solving; Distributed Constraint Optimization; DCOP

ACM Reference Format:

Khoi Hoang. 2019. Proactive Distributed Constraint Optimization Problems. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 3 pages.

1 INTRODUCTION

Distributed Constraint Optimization Problems (*DCOPs*) are problems where agents need to coordinate their value assignments to maximize the sum of the resulting constraint utilities [11, 17]. The model represents a powerful approach to the description and solution of many practical problems, serving several applications such as distributed scheduling, coordination of unmanned air vehicles, smart grid electricity networks, and sensor networks [1, 2, 4, 6, 8–10, 14, 15, 18]. In many distributed problems of interest, agents interact in complex, uncertain, and dynamic environments. For example, in distributed meeting scheduling, participants could change their preferences and priorities over time. In disaster management, new information (e.g., weather forecasts, priorities on buildings to evacuate) typically becomes available in an incremental manner. Thus, the information flow modifies the environment over time. Unfortunately, the classical DCOP paradigm is unable to model problems that change over time.

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.

Consequently, researchers have introduced *Dynamic DCOPs* (*D-DCOPs*) [7, 12, 13, 16], where utility functions can change during the problem solving process. These models make the common assumption that information on how the problem might change is unavailable. As such, existing approaches *react* to the changes in the problem and solve the current problem at hand. However, in several applications, the information on how the problem might change is indeed available, or predictable, within some degree of uncertainty.

Therefore, we introduce (finite-horizon) *Proactive Dynamic DCOPs* (*PD-DCOPs*) [3] and *Infinite-Horizon PD-DCOPs* (*IPD-DCOPs*) [5] which explicitly model how the DCOP will change over time; (*ii*) we discuss the complexity of this new class of DCOPs; and (*iii*) we develop exact and approximation algorithms with quality guarantees to solve PD-DCOPs and IPD-DCOPs *proactively*.

2 PROPOSED MODELS

2.1 PD-DCOPs

A *Proactive Dynamic DCOP* (*PD-DCOP*) is a tuple $\langle \mathbf{A}, \mathbf{X}, \mathbf{D}, \mathbf{F}, h, \mathbf{T}, c, \gamma, p^0_{\mathbf{Y}}, \alpha \rangle$, where:

- $\mathbf{A} = \{a_i\}_{i=1}^{p}$ is a set of *agents*.
- $\mathbf{X} = \{x_i\}_{i=1}^{n}$ is a mixed set of *decision* and *random variables*. To differentiate between decision variables and random variables, we use $\mathbf{Y} \subseteq \mathbf{X}$ to denote the set of random variables that model uncontrollable stochastic events.
- $\mathbf{D} = \{D_x\}_{x \in \mathbf{X}}$ is a set of finite *domains*. Each variable $x \in \mathbf{X}$ takes values from the set $D_x \in \mathbf{D}$. We also use $\Omega = \{\Omega_y\}_{y \in \mathbf{Y}} \subseteq \mathbf{D}$ to denote the set of event spaces for the random variables such that each $y \in \mathbf{Y}$ takes values in Ω_y.
- $\mathbf{F} = \{f_i\}_{i=1}^{m}$ is a set of *reward functions*, each defined over a mixed set of decision variables and random variables: $f_i : \prod_{x \in \mathbf{x}^{f_i}} D_x \to \mathbb{R}^+ \cup \{\perp\}$, where $\mathbf{x}^{f_i} \subseteq \mathbf{X}$ is *scope* of f_i and $\perp$ is a special element used to denote that a given combination of values for the variables in $\mathbf{x}^{f_i}$ is not allowed.
- $h \in \mathbb{N}$ is a finite *horizon* in which the agents can change the values of their variables.
- $\mathbf{T} = \{T_y\}_{y \in \mathbf{Y}}$ is the set of *transition functions* $T_y : \Omega_y \times \Omega_y \to [0, 1] \subseteq \mathbb{R}$ for the random variables $y \in \mathbf{Y}$, describing the probability for a random variable to change its value in successive time steps. For a time step $t > 0$, and values $\omega_i \in \Omega_y, \omega_j \in \Omega_y$, $T_y(\omega_i, \omega_j) = P(y^t = \omega_j \,|\, y^{t-1} = \omega_i)$, where y^t denotes the value of the variable y at time step t, and P is a probability measure. Thus, $T_y(\omega_i, \omega_j)$ describes the probability for the random variable y to change its value from ω_i at a time step $t-1$ to ω_j at a time step t. Finally, $\sum_{\omega_j \in \Omega_y} T_y(\omega_i, \omega_j) = 1$ for all $\omega_i \in \Omega_y$.

- $c \in \mathbb{R}^+$ is a *switching cost*, which is the cost associated with the change in the value of a decision variable between time steps.
- $\gamma \in [0, 1)$ is a *discount factor*, which represents the decrease in the importance of rewards/costs over time.
- $p^0_{\mathbf{Y}} = \{p^0_y\}_{y \in \mathbf{Y}}$ is a set of initial *probability distributions* for the random variables $y \in \mathbf{Y}$.
- $\alpha : \mathbf{X} \backslash \mathbf{Y} \rightarrow \mathbf{A}$ is a function that associates each decision variable to one agent. We assume that the random variables are not under the control of the agents and are independent of decision variables.

The goal of a PD-DCOP is to find a sequence of $h+1$ assignments $\mathbf{x}^*$ for all the decision variables in $\mathbf{X} \setminus \mathbf{Y}$:

$$\mathbf{x}^* = \underset{\mathbf{x}=\langle \mathbf{x}^0, \ldots, \mathbf{x}^h \rangle \in \Sigma^{h+1}}{\operatorname{argmax}} \mathcal{F}^h(\mathbf{x}) \tag{1}$$

$$\mathcal{F}^h(\mathbf{x}) = \sum_{t=0}^{h-1} \gamma^t \left[\mathcal{F}_x^t(\mathbf{x}^t) + \mathcal{F}_y^t(\mathbf{x}^t) \right] \tag{2}$$

$$- \sum_{t=0}^{h-1} \gamma^t \left[c \cdot \Delta(\mathbf{x}^t, \mathbf{x}^{t+1}) \right] \tag{3}$$

$$+ \tilde{\mathcal{F}}_x(\mathbf{x}^h) + \tilde{\mathcal{F}}_y(\mathbf{x}^h) \tag{4}$$

where Σ is the assignment space for the decision variables of the PD-DCOP, at each time step. These assignments to values of decision variables maximizes the sum of two terms. The first term maximizes the discounted net utility, that is, the discounted rewards for the functions that do not involve exogenous factors ($\mathcal{F}_x$) and the expected discounted random rewards ($\mathcal{F}_y$) minus the discounted penalties over the first h time steps. The second term maximizes the discounted future rewards for the problem.

2.2 IPD-DCOPs

At a high level, the *Infinite-Horizon Proactive Dynamic DCOP* (IPD-DCOP) model is a straightforward, but essential and significant, extension of the PD-DCOP model. Both IPD-DCOP and PD-DCOP models assume that a finite horizon h is given. However, PD-DCOPs ignore all changes to the problem after its finite horizon h and, thus, do not optimize for them, while IPD-DCOPs don't take into account the discount factor and assume that the Markov chains will converge to stationary distributions after that horizon and an optimal solution for those stationary distributions should be adopted after the horizon.

As agents will keep their optimal solution from horizon h onwards, the goal of an IPD-DCOP is to find a sequence of $h+1$ assignments $\bar{\mathbf{x}}^*$ for all the decision variables in $\mathbf{X}$:

$$\bar{\mathbf{x}}^* = \underset{\bar{\mathbf{x}}=\langle \mathbf{x}^0, \ldots, \mathbf{x}^h \rangle \in \Sigma^{h+1}}{\operatorname{argmax}} \mathcal{F}(\bar{\mathbf{x}}) \tag{5}$$

$$\mathcal{F}(\bar{\mathbf{x}}) = \sum_{t=0}^{h} \mathcal{F}^t(\mathbf{x}^t) \tag{6}$$

$$- \sum_{t=0}^{h-1} \left[c \cdot \Delta(\mathbf{x}^t, \mathbf{x}^{t+1}) \right] \tag{7}$$

where Σ is the assignment space for the decision variables of the IPD-DCOP at each time step, $\Delta : \Sigma \times \Sigma \rightarrow \{0\} \cup \mathbb{N}$ is a function counting the number of assignments to decision variables that differs from one time step to the next.

Note that at horizon h, agents solve the problem with the stationary distribution of random variables and keep this optimal solution onwards. This would maximize the expected reward at each time step after the distributions have converged to the stationary distribution. If choosing any other assignment, then, it will result in a smaller reward.

3 CONCLUSIONS

In real-world applications, agents often act in dynamic environments. Thus, the Dynamic DCOP formulation is attractive to model such problems. Current research has focused at solving such problems *reactively*, thus discarding the information on possible future changes, which is often available in many applications. To cope with this limitation, we (*i*) introduce *Proactive Dynamic DCOPs* (*PD-DCOPs*), which model the dynamism in Dynamic DCOPs; (*ii*) provide theoretical results on the complexity class of PD-DCOPs; and (*iii*) develop an exact PD-DCOP algorithm that solves the problem *proactively* as well as an approximation algorithm with quality guarantees that can scale to larger and more complex problems. Moreover, we also propose the Infinite-Horizon PD-DCOP (IPD-DCOP) model, which extends PD-DCOPs to optimize the cumulative reward obtained across an infinite number of time steps. It exploits the convergence properties of Markov chains and assumes that the underlying Markov chain in the problem is guaranteed to converge to the unique stationary distribution.

ACKNOWLEDGEMENTS

This work is supervised by Dr. William Yeoh at Washington University in St. Louis, and supported by NSF grants 1345232, 1540168, 1550662 and JSPS KAKENHI Grant Number 24220003.

REFERENCES

[1] Francesco Delle Fave, Alex Rogers, Zhe Xu, Salah Sukkarieh, and Nicholas Jennings. 2012. Deploying the Max-Sum Algorithm for Decentralised Coordination and Task Allocation of Unmanned Aerial Vehicles for Live Aerial Imagery Collection. In *Proceedings of ICRA*. 469–476.

[2] Alessandro Farinelli, Alex Rogers, and Nicholas Jennings. 2014. Agent-Based Decentralised Coordination for Sensor Networks using the Max-Sum Algorithm. *Autonomous Agents and Multi-Agent Systems* 28, 3 (2014), 337–380.

[3] Khoi D. Hoang, Ferdinando Fioretto, Ping Hou, Makoto Yokoo, William Yeoh, and Roie Zivan. 2016. Proactive Dynamic Distributed Constraint Optimization. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 597–605.

[4] Khoi D. Hoang, Ferdinando Fioretto, William Yeoh, Enrico Pontelli, and Roie Zivan. 2018. A Large Neighboring Search Schema for Multi-Agent Optimization. In *Proceedings of the International Conference on Principles and Practice of Constraint Programming (CP)*. 688–706.

[5] Khoi D. Hoang, Ping Hou, Ferdinando Fioretto, William Yeoh, Roie Zivan, and Makoto Yokoo. 2017. Infinite-Horizon Proactive Dynamic DCOPs. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 212–220.

[6] Akshat Kumar, Boi Faltings, and Adrian Petcu. 2009. Distributed constraint optimization with structured resource constraints. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. International Foundation for Autonomous Agents and Multiagent Systems, 923–930.

[7] Robert Lass, Evan Sultanik, and William Regli. 2008. Dynamic Distributed Constraint Reasoning. In *Proceedings of AAAI*. 1466–1469.

[8] Thomas Léauté and Boi Faltings. 2011. Coordinating Logistics Operations with Privacy Guarantees. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. 2482–2487.

[9] Rajiv Maheswaran, Milind Tambe, Emma Bowring, Jonathan Pearce, and Pradeep Varakantham. 2004. Taking DCOP to the Real World: Efficient Complete Solutions for Distributed Event Scheduling. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 310–317.

[10] Sam Miller, Sarvapali Ramchurn, and Alex Rogers. 2012. Optimal Decentralised Dispatch of Embedded Generation in the Smart Grid. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 281–288.

[11] Pragnesh Modi, Wei-Min Shen, Milind Tambe, and Makoto Yokoo. 2005. ADOPT: Asynchronous Distributed Constraint Optimization with Quality Guarantees. *Artificial Intelligence* 161, 1–2 (2005), 149–180.

[12] Adrian Petcu and Boi Faltings. 2005. Superstabilizing, Fault-Containing Multiagent Combinatorial Optimization. In *Proceedings of AAAI*. 449–454.

[13] Adrian Petcu and Boi Faltings. 2007. Optimal Solution Stability in Dynamic, Distributed Constraint Optimization. In *Proceedings of IAT*. 321–327.

[14] Suguru Ueda, Atsushi Iwasaki, and Makoto Yokoo. 2010. Coalition Structure Generation Based on Distributed Constraint Optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 197–203.

[15] William Yeoh, Ariel Felner, and Sven Koenig. 2010. BnB-ADOPT: An Asynchronous Branch-and-Bound DCOP Algorithm. *Journal of Artificial Intelligence Research* 38 (2010), 85–133.

[16] William Yeoh, Pradeep Varakantham, Xiaoxun Sun, and Sven Koenig. 2015. Incremental DCOP Search Algorithms for Solving Dynamic DCOPs. In *Proceedings of IAT*. 257–264.

[17] William Yeoh and Makoto Yokoo. 2012. Distributed Problem Solving. *AI Magazine* 33, 3 (2012), 53–65.

[18] Roie Zivan, Harel Yedidsion, Steven Okamoto, Robin Glinton, and Katia Sycara. 2015. Distributed Constraint Optimization for Teams of Mobile Sensing Agents. *Journal of Autonomous Agents and Multi-Agent Systems* 29, 3 (2015), 495–536.